The background features several geometric shapes: a cyan outline of a house-like shape on the left, a large green chevron pointing right on the right side, and a dark green chevron pointing left at the bottom. A green double-line border encloses the central text.

# **COLABORACIÓN ACCENTURE - FPs**

# AGENDA

- Bienvenida
- Agile y Scrum
- Ejemplo práctico
- Debate
- Cierre

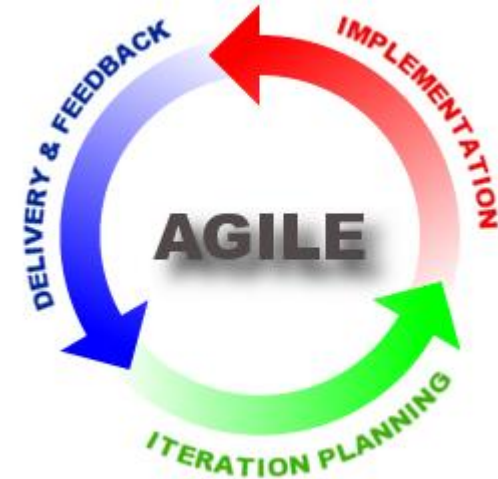
# AGENDA

- Bienvenida
- Agile y Scrum
- Ejemplo práctico
- Debate
- Cierre

# ¿QUÉ ES AGILE?

Es un término que engloba todas aquellas metodologías que promueven la gestión y desarrollos de proyectos de forma **iterativa, incremental**, con una frecuente **inspección** y **adaptación**.

No se trata de un nuevo proceso, es un **cambio cultural** en el desarrollo de productos



# CAMBIO CULTURAL

## COMPORTAMIENTOS Y ARTEFACTOS

Sólo lo que vemos

## PRÁCTICAS

## MODELOS MENTALES

Cómo racionalizamos  
Estructuras cognitivas

## PRINCIPIOS

## VALORES Y CREENCIAS

En lo que creemos  
Educación  
Lo que es correcto

## VALORES



# AGILE - VALORES

Individuos e interacciones

**vs.**

Procesos y herramientas

Incremento de producto

**vs.**

Documentación comprensible

Colaboración con el cliente

**vs.**

Contrato negociado

Responder al cambio

**vs.**

Seguir un plan

# AGILE - PRINCIPIOS

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

# AGILE - PRINCIPIOS

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



# LEYENDAS URBANAS

01

En **AGILE** no se estima

**AGILE** no tiene encaje  
en proyectos cerrados

02

03

**AGILE** no se puede  
utilizar en proyectos Off-  
Shore

**AGILE** vale para cualquier  
tipo de proyecto

04

# REALIDADES

01

Es imposible recoger todos los requisitos al principio de un proyecto

Cualquier requisito recogido **cambiará**

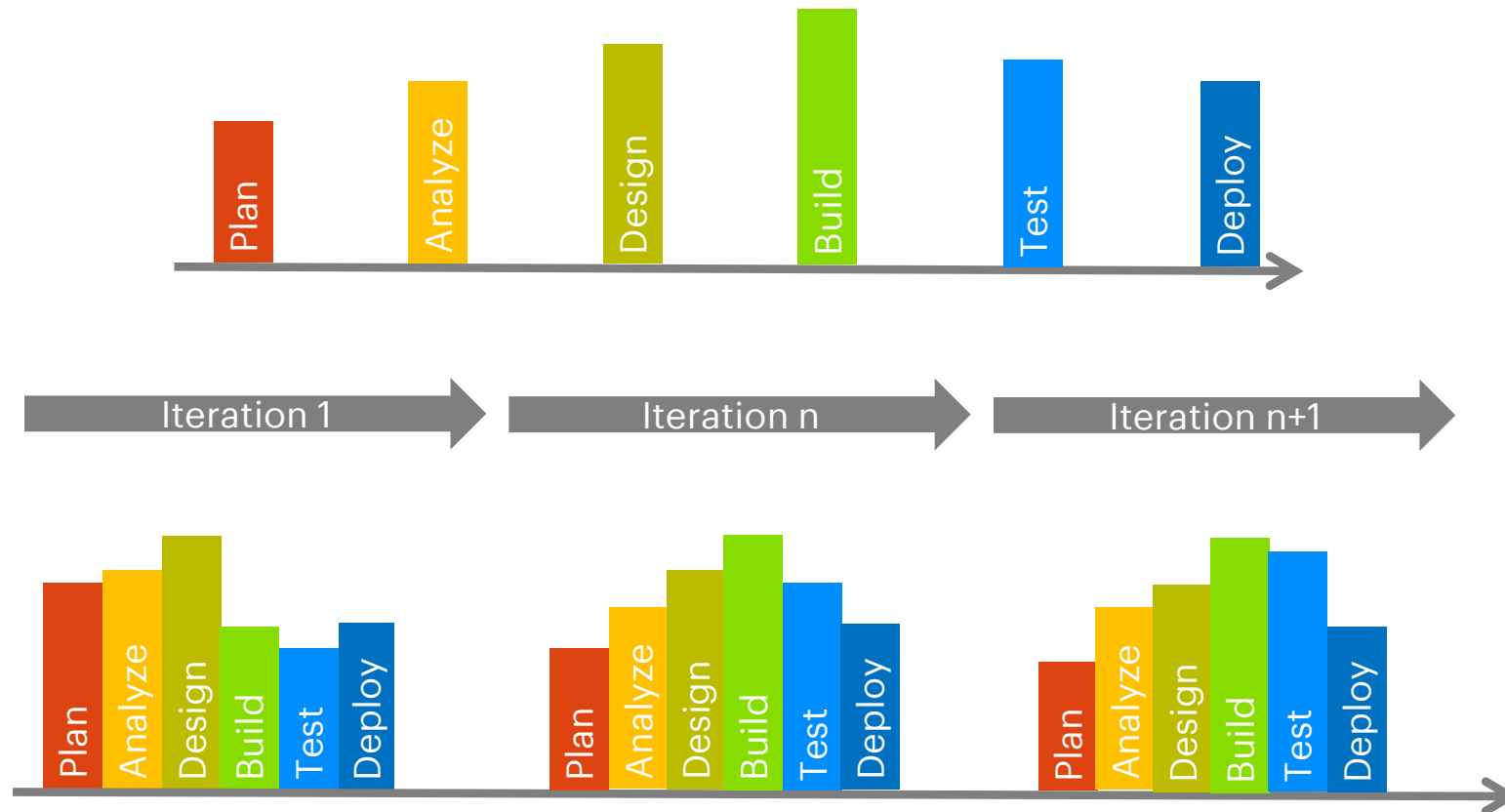
02

03

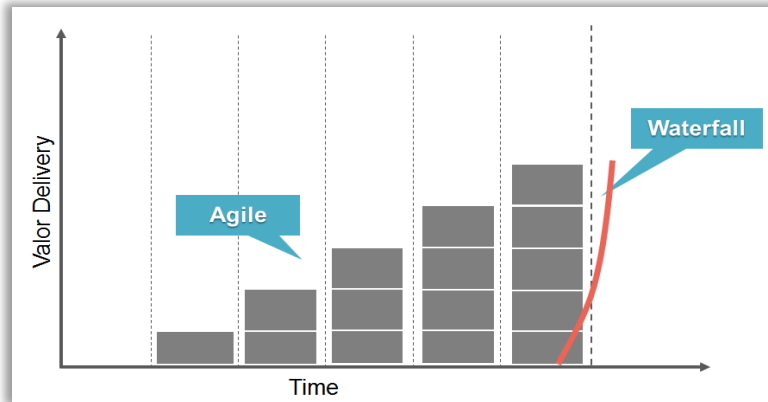
Siempre querremos hacer más de lo que el dinero y el tiempo permita

# AGILE VS. TRADITIONAL APPROACH

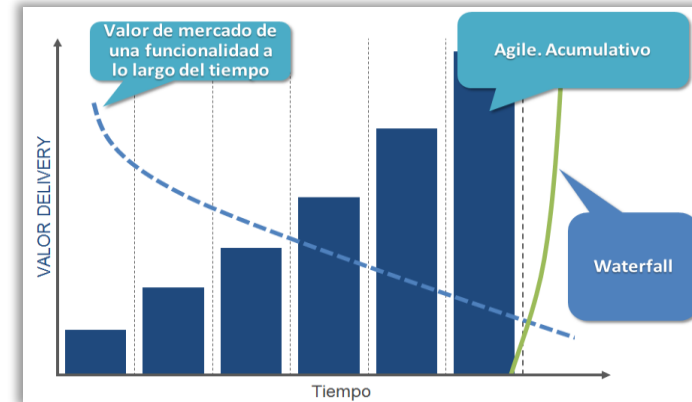
Agile Development is focused on an **iterative** (addressing all aspects of the lifecycle in each iteration), **incremental** and **flexible** approach to software development



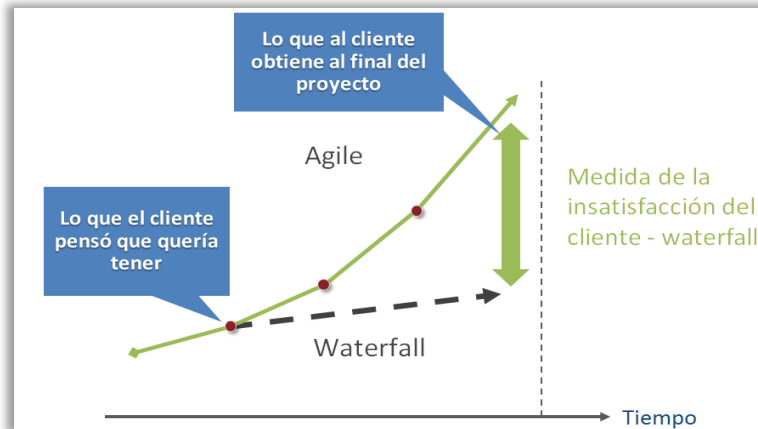
# BENEFICIOS



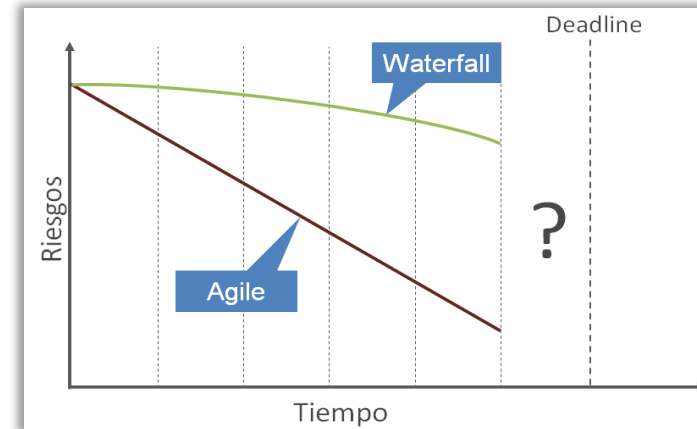
**Entrega del producto antes**



**Time to Market**

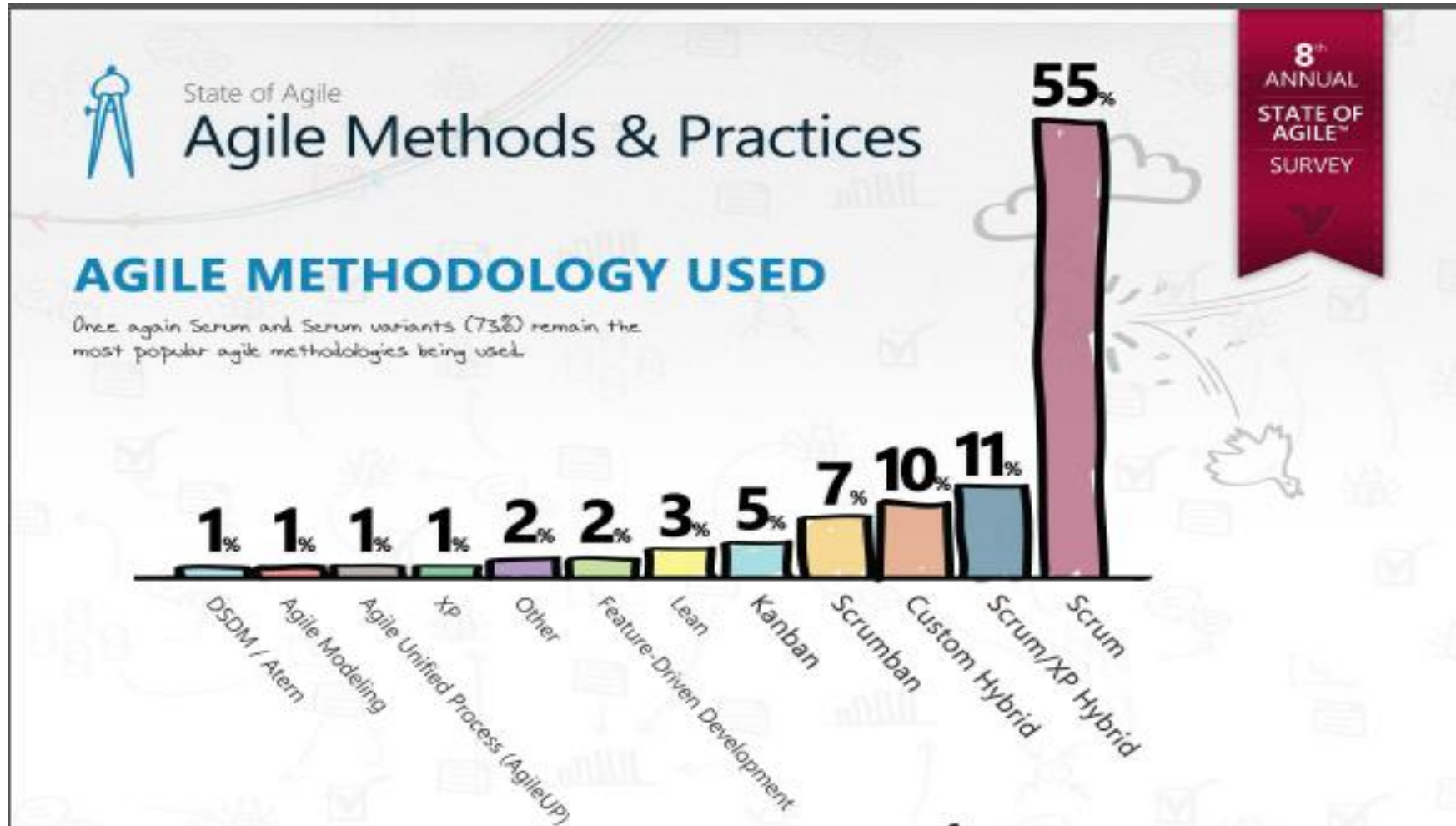


**Adaptabilidad**



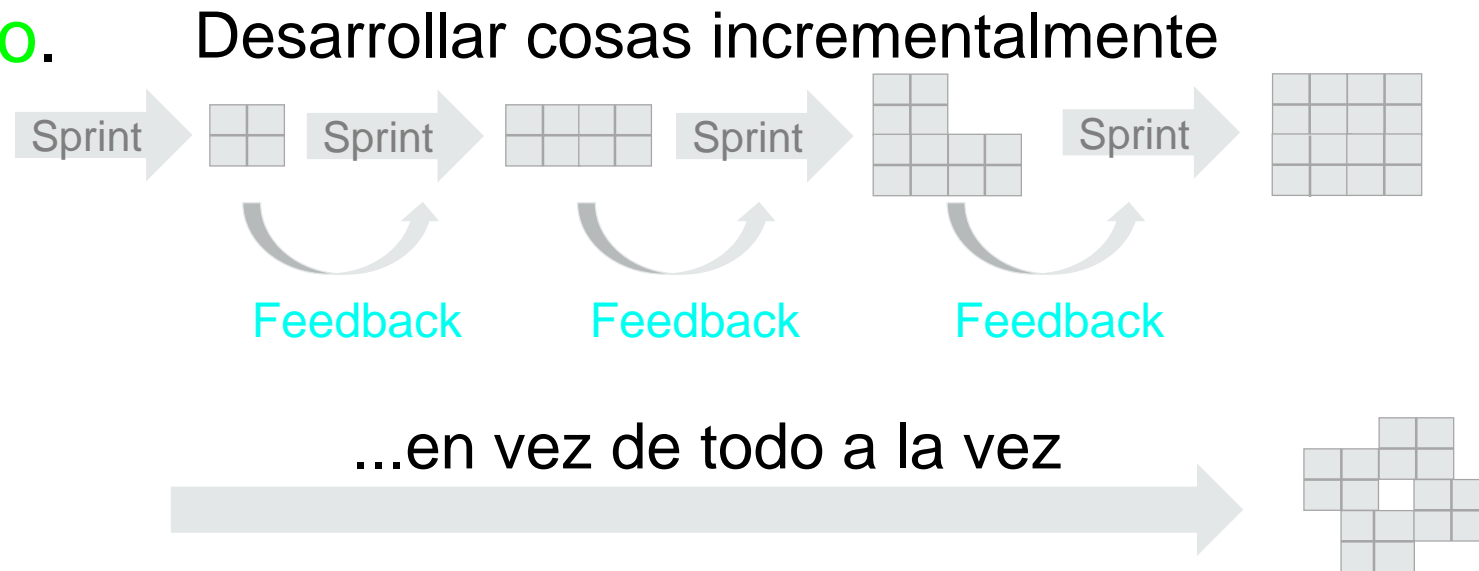
**Identificación temprana de riesgos**

# METODOLOGÍAS AGILE

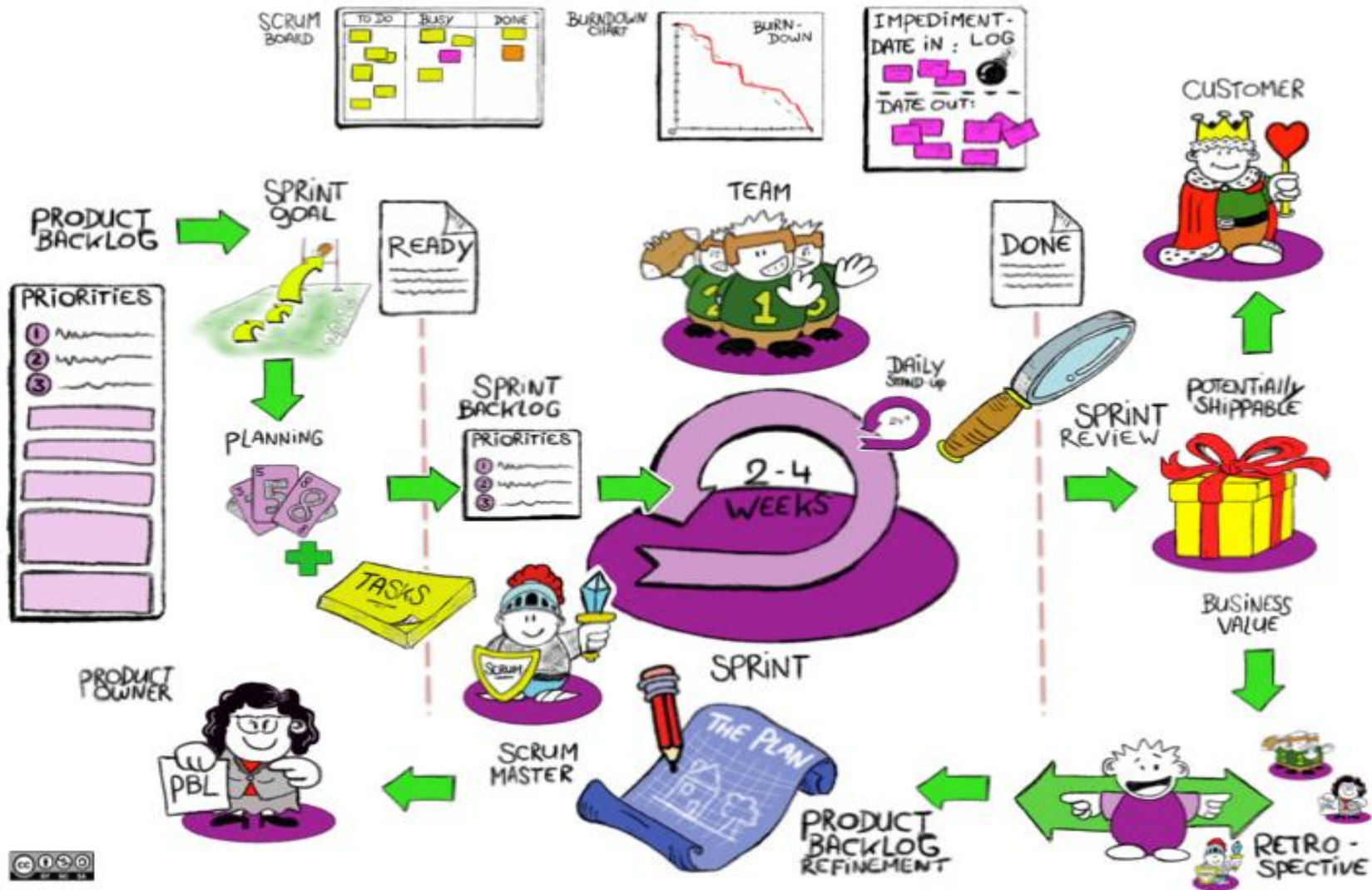


# SCRUM

Scrum es un conjunto de **principios y prácticas** que ayudan a los equipos a desarrollar productos en **periodos cortos (llamados sprints)**, permitiendo **feedback rápido, mejora continua** y una rápida **adaptación al cambio**.

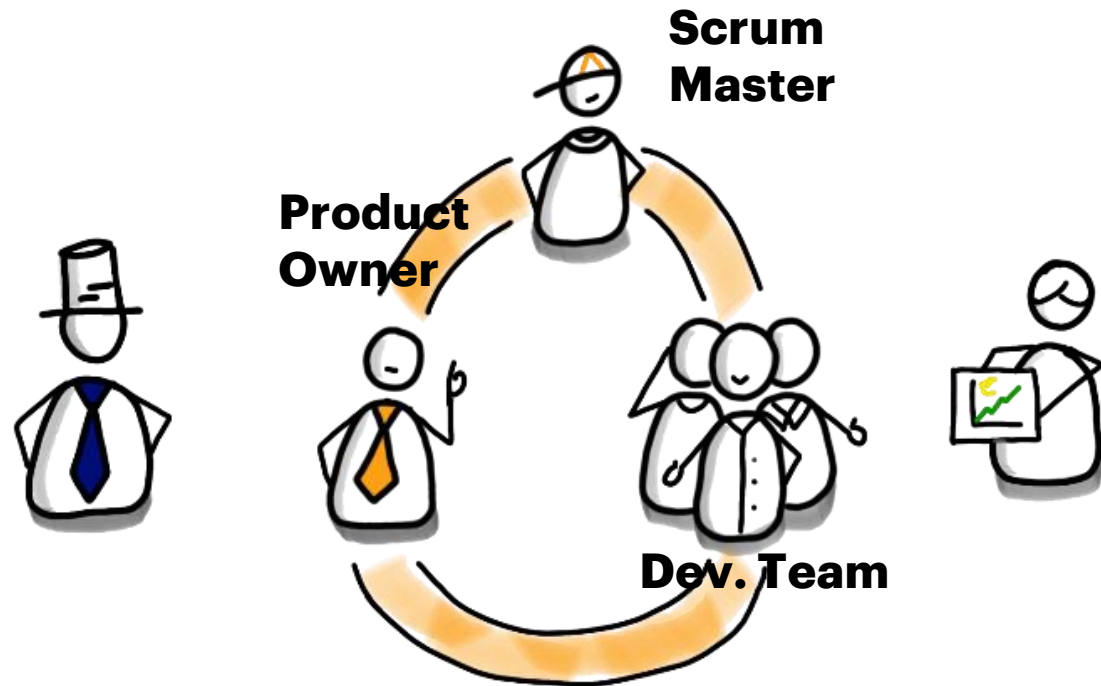


# SCRUM



# SCRUM – ROLES

- Product Owner
- Scrum Master
- Team





# SCRUM – ROLES – PRODUCT OWNER

- La persona más cercana a negocio/cliente.
- Aglutina los intereses de los stakeholders.
- Única persona responsable del Retorno de Inversión (ROI).
- Responsable de la visión del producto.
- Reprioriza y modifica el Product Backlog en cualquier momento.
- Responsable final sobre las preguntas del requerimiento/funcionalidad.
- Aprueba o rechaza el incremento de producto.



# SCRUM – ROLES – DEV. TEAM

- Multifuncional.
- Auto-organizado y auto-gestionado, sin roles puestos desde fuera.
- Decide la cantidad de trabajo que es capaz de abordar en un sprint, negociando con el PO.
- Autónomo en cuanto a la forma de lograr compromisos.
- Mayor éxito si están ubicados en el mismo sitio y trabajan a full time en el proyecto.
- Número ideal: 7 +/- 2 personas



# SCRUM – ROLES – SCRUM MASTER

- Encargado de evangelizar SCRUM tanto dentro del equipo como fuera.
- Ayuda a resolver impedimentos.
- Crea un ambiente para la auto organización.
- Protege al equipo de interferencias externas.
- Facilita las reuniones SCRUM.
- Recolecta métricas.
- **No gestiona**, no es una figura de autoridad.



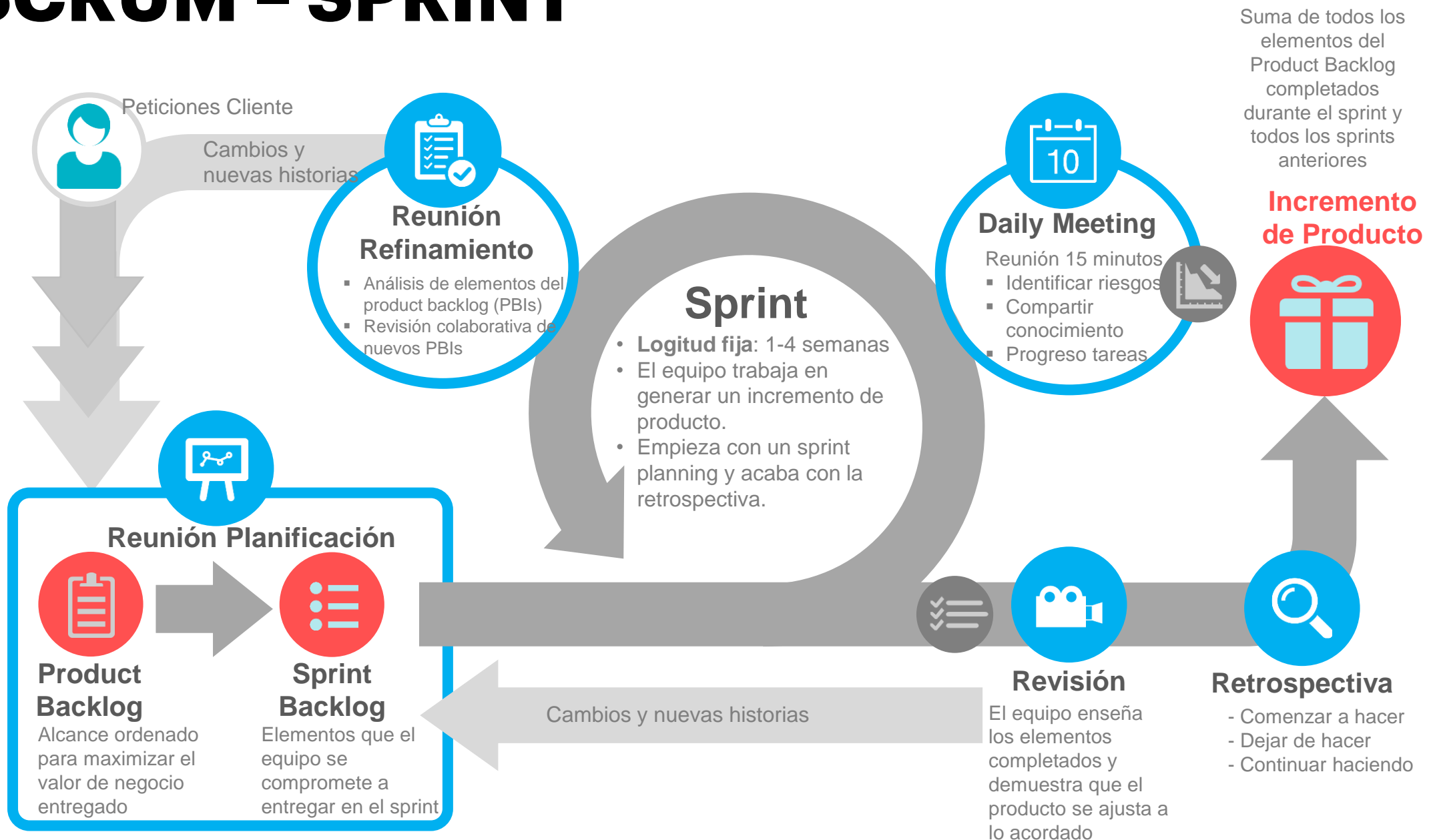
# SCRUM – ARTEFACTOS

- Product Backlog
- Sprint Backlog
- The Burn-down chart
- The Impediment Log

# SCRUM – CEREMONIAS

- Daily
- Sprint Planning
- Sprint Retrospective
- Spring Review
- Refinement

# SCRUM - SPRINT



# INCEPTION (SPRINT 0)

- No es una ceremonia oficial de Scrum.
- No tiene una longitud predefinida.
- Se aplica antes de empezar el ciclo de vida Scrum.
- Puede verse como un set up previo:
  - Definición del equipo.
  - Presupuesto.
  - Definición arquitectura (demo).
  - Estimación inicial.
  - Definición de DoR y DoD oficial.
  - Refinamiento inicial.

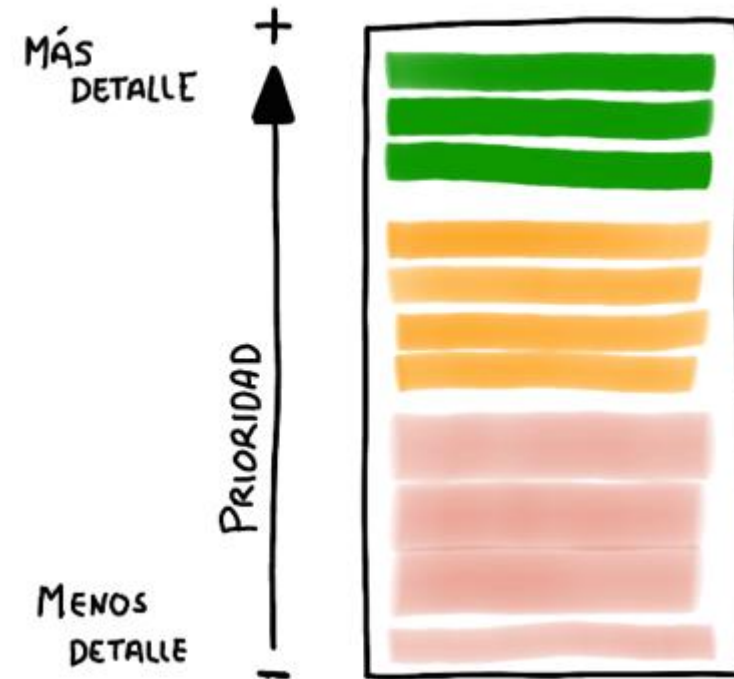


# SCRUM - PRODUCT BACKLOG

- Lista ordenada y priorizada de funcionalidades proporcionada por el cliente (MosCoW)
- Visible por todo el mundo (Equipo y Stakeholder)

MoSCoW:

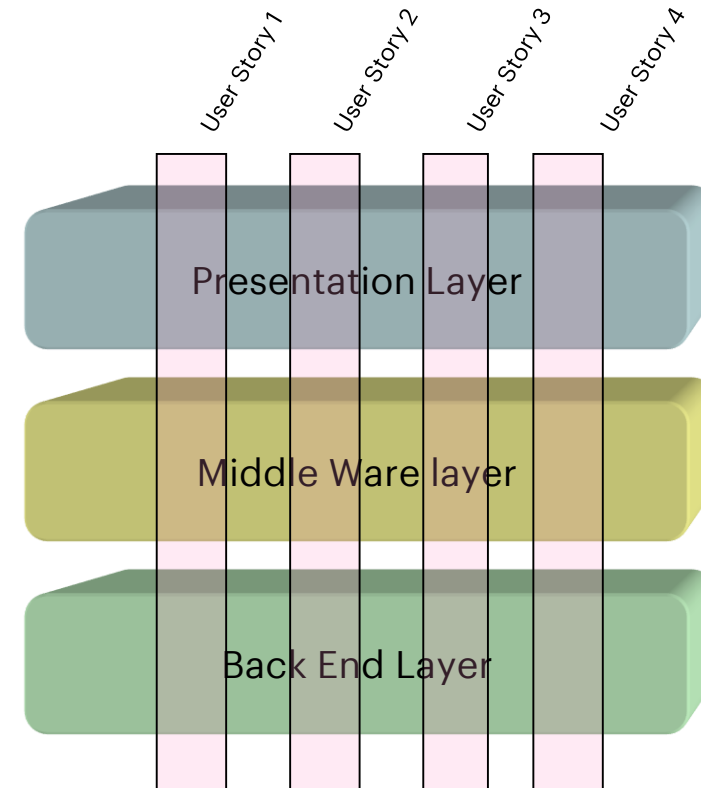
- Must have (or Minimum Usable Subset)
  - Should have
  - Could have
  - Won't have (but Would like in future)
- Combinación de épicas e historias de usuario.
  - EL PO es el propietario y el encargado de priorizarlo.
  - Los items puede ser añadidos o modificados en cualquier momento, siempre y cuando no afecten al Sprint en curso





# SCRUM – HISTORIA DE USUARIO

- Una historia de usuario (item del product backlog) es una característica o funcionalidad que tiene valor para el usuario final.
- Lo suficientemente pequeña para que pueda ser abordada dentro de un Sprint.
- Suele tener el formato:
  - Yo, como <rol>, quiero <funcionalidad> para <motivo>
  - Criterio de aceptación
  - Prioridad
  - Estimación



# SCRUM – CRITERIO INVEST

<b>I</b> Independent	<b>I</b> Independent <ul style="list-style-type: none"><li>❖ Dependencies lead to problems estimating and prioritizing</li><li>❖ Can ideally select a Story to work on without pulling in 18 other stories</li></ul>
<b>N</b> Negotiable	<b>N</b> Negotiable <ul style="list-style-type: none"><li>❖ Stories are not contracts</li><li>❖ Leave or imply some flexibility</li></ul>
<b>V</b> Valuable	<b>V</b> Valuable <ul style="list-style-type: none"><li>❖ To users or customers, not developers</li><li>❖ Re-write developer stories to reflect value to users or customers</li></ul>
<b>E</b> Estimatable	<b>E</b> Estimatable <p>Because plans are based on User Stories, we need to be able to estimate them</p>
<b>S</b> Sized appropriately	<b>S</b> Sized Appropriately <ul style="list-style-type: none"><li>❖ Small enough to complete in one sprint if you're about to work on it</li><li>❖ Bigger if further off on the horizon</li></ul>
<b>T</b> Testable	<b>T</b> Testable <ul style="list-style-type: none"><li>❖ Testable so that you have an easy, binary way of knowing whether a story is finished – “Definition of Done”</li><li>❖ Done or not done: no “partially finished” or “done except”</li></ul>

# SCRUM – FORMATO HISTORIA USUARIO

**Story Title :**

**Release & Sprint # :**

As a **<User of the Application>**, I want to **< Required Functionality >** so that,  
**<Reason for the functionality>**.

- **Priority :** Must Have/Should Have/Could Have/Wont Have
- **Estimate :** X Story Points
- **Business Value (Optional) :**
- **Owner :**

**Acceptance Criteria:**

.....

.....

# SCRUM – FORMATO HISTORIA USUARIO

**Dependent User Stories:**

**Technical Risks :**

**User Story Test / Design Considerations:**

# SCRUM – STORY POINTS

- Unidad de medida que expresa el tamaño de una historia de usuario.
- Cada historia de usuario del product backlog debe ser estimada.
- El criterio seguido para establecer el tamaño de una historia es:
  - Tamaño: Esfuerzo + Complejidad + Incertidumbre
    - Esfuerzo: Cuánto tiempo nos llevará.
    - Complejidad: Cómo de difícil es.
    - Incertidumbre: Grado de inseguridad para su implementación.

# SCRUM - ESTIMACIÓN



Cada miembro dispone de un mazo de cartas basado en la serie Fibonacci: 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 60, 100 ?

Las historias de usuario son explicadas por el cliente/PO. El equipo Desarrollo realiza todas las preguntas necesarias para su estimación.

Cada miembro del equipo de Desarrollo selecciona una carta, sin mostrársela al resto.

Las cartas son levantadas a la vez.

Se argumentan las diferencias, sobre todo las estimaciones más extremas.

Se re-estima hasta que se consigue consenso



# AGENDA

- Bienvenida
- Agile y Scrum
- Ejemplo práctico
- Debate
- Cierre

# AGENDA

- Bienvenida
- Agile y Scrum
- Ejemplo práctico
- Debate
- Cierre



# AGENDA

- Bienvenida
- Agile y Scrum
- Ejemplo práctico
- Debate
- Cierre